Math 116: Problem Set 2

Owen Jones

1/21/2024

1. $x_{n+3} = c_0 x_n + c_1 x_{n+1} + c_2 x_{n+2}$
   $\Rightarrow \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$
   $\Rightarrow (c_0, c_1, c_2) = (1, 0, 1)$
   $\Rightarrow x_{n+3} = x_n + x_{n+2}$
   $\Rightarrow$ 1001 are the next 4 elements of the sequence.

2. $\det(M_3) = \det \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} = 1 + 0 - 1 = 0 \mod 2$
   $\Rightarrow \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
   $\Rightarrow (c_0, c_1) = (1, 0)$
   $\Rightarrow x_{n+2} = x_n$

3. (a) If Eve observes the ciphertext repeats with a period of 6, she can deduce that the plaintext is one repeated letter and the key is length 6. Eve knows every $n-th$ character will be shifted by the same amount, and she notices every $6th$ letter is the same. If the key is of length 6, then every $6th$ letter is the same letter. Since each of the congruence classes $\mod 6$ are shifted by different amounts, a good guess is to assume that every character is the same letter.

   (b) Using the property that no 6 letter word is a shift of another word, the fastest way to determine the key is by brute force. Shift the first 6 characters of the ciphertext by $1 \mod 26$ until an English word is obtained.

   (c) $\# \text{ of matches} = \begin{cases} \text{length of ciphertext} - n & \text{for } n \equiv 0 \mod 6 \\ 0 & \text{for } n \not\equiv 0 \mod 6 \end{cases}$
   where n is the number of displacements.

4. The message is EVEISEAVESDROPPINGONUS

5. The key is JACK and the message is WEUSEWORDSLIKEHONORCODELOYALTYWEUSETHESEWORDSASTHEBACKBONE OFALIFE-

SPENTDEFENDINGSOMETHINGYOUUSETHEMASAPUNCHLINEIHAVENEITHERTHETIMENORTHEINCLINATIONTOEXPLAIN MYSELFTOAMANWHORISESANDSLEEPSUNDERTHEBLANKETOFTHEVERYFREEDOMTHATIPROVIDEANDTHENQUESTIONSTHEMANNER INWHICHIPROVIDEIT

6. The key is WATSON and the message is 'HOLMESHADBEENSEATEDFORSOMEHOURSINSILENCEWITHHISLONG THINBACKCURVEDOVERACHEMICALVESSELINWHICHHEWASBREWINGAPARTICULARLYMALODOROUSPRODUCTH
   UPONHISBREASTANDHELOOKEDFROMMYPOINTOFVIEWLIKEASTRANGELANKBIRDWITHDULLGREYPLUMAGEANDABLACKTOPKNOT SOWATSONSAIDHESUDDENLYYOUDONOTPROPOSETOINVESTINSOUTHAFRICANSECURITIES'

In [1]:

```
import math
import numpy as np
import pandas as pd
```

In [2]:

```
letter_frequencies = {
    "A": 0.082, "B": 0.015, "C": 0.028, "D": 0.043, "E": 0.127, "F": 0.022,
    "G": 0.020, "H": 0.061, "I": 0.070, "J": 0.002, "K": 0.008, "L": 0.040,
    "M": 0.024, "N": 0.067, "O": 0.075, "P": 0.019, "Q": 0.001, "R": 0.060,
    "S": 0.063, "T": 0.091, "U": 0.028, "V": 0.010, "W": 0.023, "X": 0.001,
    "Y": 0.020, "Z": 0.001,
}
```

In [3]:

```
ciphertext_1='ZDVOGZIMKGYZFVDDVXUBPA'
```

In [118…

```
ciphertext_2='''FEWCNWQBMSNSTEJYWOTMXDGVXYCVCYYODSGDQEUOFOTNBAUDQEDKLKDYWEQPJLKF
```

In [157…

```
ciphertext_3='DOEESFDAWTSRJSXSHRZFHJGBIEAGIEOIGKWYANVWKVPHAAGYKNZLVVJBTUYPQROWRE
```

In [75]:

```
def frequency_calculator(text):
    frequencies=dict()
    for letter in text:
        if letter not in frequencies.keys():
            frequencies[letter]=1
        else:
            frequencies[letter]+=1
    for letter in frequencies.keys():
        frequencies[letter]=frequencies[letter]/len(text)
    return dict(sorted(frequencies.items()))
```

In [32]:

```
def tonum(char):
    "Converts a letter of the alphabet into a number in the range 0..25"
    return ord(char) - 65 # 65 is the ASCII code for the letter A
def tochar(num):
    "Converts a number in the range 0..25 into a letter of the alphabet"
    return chr(num + 65) # 65 is the ASCII code for the letter A
```

In [121…

```
def vigenere_decrypt(text,key):
    key=key.upper()
    length=len(key)
    decrypted=[]
    for index,letter in enumerate(text):
        decrypted.append(tochar((tonum(letter)-tonum(key[index%length]))%26))
    return ''.join(decrypted)
```

In [134…

```
vigenere_decrypt(ciphertext_1,'VIRGO')
```

Out[134…

```
'EVEISEAVESDROPPINGONUS'
```

In [138…

```
def vigenere_key(text,length):
    key_array=[]
    for i in range(length):
        dot_product=[]
        frequencies=frequency_calculator(text[i::length])
        for j in range(26):
            shifted=dict()
            for key in letter_frequencies.keys():
                shifted[tochar((tonum(key)+j)%26)]=letter_frequencies[key]
            dot_product.append(sum(shifted[key]*frequencies.get(key,0) for key i
        key_array.append(tochar(dot_product.index(max(dot_product))))
    return ''.join(key_array)
```

In [139…

```
vigenere_key(ciphertext_2,4)
```

```
Out[139… 'JACK'
```

```
In [140… vigenere_decrypt(ciphertext_2,'JACK')
```

```
Out[140… 'WEUSEWORDSLIKEHONORCODELOYALTYWEUSETHESEWORDSASTHEBACKBONEOFALIFESPENTDEFENDING
         SOMETHINGYOUUSETHEMASAPUNCHLINEIHAVENEITHERTHETIMENORTHEINCLINATIONTOEXPLAINMYSE
         LFTOAMANWHORISESANDSLEEPSUNDERTHEBLANKETOFTHEVERYFREEDOMTHATIPROVIDEANDTHENQUEST
         IONSTHEMANNERINWHICHIPROVIDEIT'
```

```
In [154… def vigenere_length(text,length=20):
             length_array=[]
             for i in range(1,length):
                 shifted=''.join([' ']*i+list(text))
                 length_array.append(sum(x==y for x,y in zip(text,shifted)))
             return length_array.index(max(length_array))+1
```

```
In [158… vigenere_length(ciphertext_3,length=20)
```

```
Out[158… 6
```

```
In [159… vigenere_key(ciphertext_3,6)
```

```
Out[159… 'WATSON'
```

```
In [160… vigenere_decrypt(ciphertext_3,"WATSON")
```

```
Out[160… 'HOLMESHADBEENSEATEDFORSOMEHOURSINSILENCEWITHHISLONGTHINBACKCURVEDOVERACHEMICALV
         ESSELINWHICHHEWASBREWINGAPARTICULARLYMALODOROUSPRODUCTHISHEADWASSUNKUPONHISBREAS
         TANDHELOOKEDFROMMYPOINTOFVIEWLIKEASTRANGELANKBIRDWITHDULLGREYPLUMAGEANDABLACKTOP
         KNOTSOWATSONSAIDHESUDDENLYYOUDONOTPROPOSETOINVESTINSOUTHAFRICANSECURITIES'
```

```
In [ ]:
```